

PATENT
2557-000198/US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Jang-Ho CHO
Application No.: NEW
Filed: January 12, 2004
For: MULTI-THREADED PROCESSOR AND METHOD OF
MULTI-THREADED PROCESSING

PRIORITY LETTER

January 12, 2004

COMMISSIONER FOR PATENTS
P.O. BOX 1450
Alexandria, Virginia 22313-1450

Dear Sirs:

Pursuant to the provisions of 35 U.S.C. 119, enclosed is/are a certified copy of the following priority document(s).

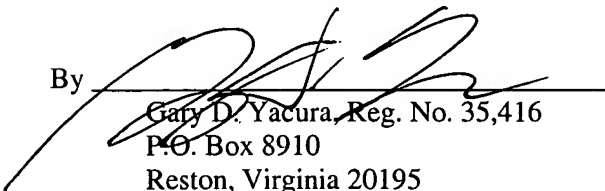
<u>Application No.</u>	<u>Date Filed</u>	<u>Country</u>
10-2003-0005236	January 27, 2003	REPUBLIC OF KOREA

In support of Applicant's priority claim, please enter this document into the file.

Respectfully submitted,

HARNESS, DICKEY, & PIERCE, P.L.C.

By


Gary D. Yacura, Reg. No. 35,416
P.O. Box 8910
Reston, Virginia 20195
(703) 668-8000

GDY:jj



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2003-0005236
Application Number

출원 년 월 일 : 2003년 01월 27일
Date of Application
JAN 27, 2003

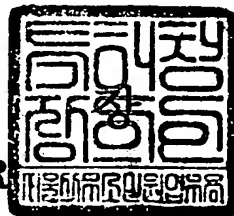
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 05 월 30 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0022
【제출일자】	2003.01.27
【국제특허분류】	G06F
【발명의 명칭】	명령어수에 수행 주기 회수를 가중치로 사용하여 쓰레드를 폐치하는 동시 다중 쓰레딩 프로세서 및 그 방법
【발명의 영문명칭】	Simultaneous Multithreading processor providing for thread fetch based on instructions count weighted by the operation cycle count and method thereof
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2003-003435-0
【대리인】	
【성명】	정상빈
【대리인코드】	9-1998-000541-1
【포괄위임등록번호】	2003-003437-4
【발명자】	
【성명의 국문표기】	조장호
【성명의 영문표기】	CHO, Jang Ho
【주민등록번호】	690416-1402723
【우편번호】	449-846
【주소】	경기도 용인시 수지읍 풍덕천리 진산마을 삼성5차@ 526-202
【국적】	KR
【심사청구】	청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인

이영필 (인) 대리인

정상빈 (인)

【수수료】

【기본출원료】

20 면 29,000 원

【가산출원료】

7 면 7,000 원

【우선권주장료】

0 건 0 원

【심사청구료】

10 항 429,000 원

【합계】

465,000 원

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

명령어수에 수행 주기 회수(cycle count)를 가중치로 사용하여 쓰레드(thread)를 페치(fetch)하는 동시 다중 쓰레딩(SMT) 프로세서 및 그 방법이 개시된다. 상기 동시 다중 쓰레딩(SMT) 프로세서는, 페치부가 명령어 수에 그 수행 주기 회수(cycle count)를 가중치로 사용하여 쓰레드(thread)를 선택하고, 그 쓰레드 중 실행될 명령어의 어드레스를 가리키고 있는 프로그램 카운터에서 어드레스를 페치(fetch)한다. 따라서, SMT의 성능을 더욱 향상시키고, 결국 CPU 등의 시스템 성능을 더욱 향상시키는 효과가 있다.

【대표도】

도 1

【명세서】**【발명의 명칭】**

명령어수에 수행 주기 회수를 가중치로 사용하여 쓰레드를 페치하는 동시 다중 쓰레딩 프로세서 및 그 방법{Simultaneous Multithreading processor providing for thread fetch based on instructions count weighted by the operation cycle count and method thereof}

【도면의 간단한 설명】

본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 간단한 설명이 제공된다.

도 1은 본 발명의 일실시예에 따른 동시 다중 쓰레딩(SMT) 프로세서의 블록도이다.

도 2는 도 1의 페치부(fetch unit)(110)의 구체적인 블록도이다.

도 3은 도 2의 동시 다중 쓰레딩(SMT) 프로세서의 동작 설명을 위한 흐름도이다.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<5> 본 발명은 컴퓨터 시스템의 CPU(중앙연산처리장치)에 관한 것으로, 특히 컴퓨터 시스템의 동시 다중 쓰레딩(Simultaneous Multithreading)(이하 "SMT"라 약칭함) 프로세서에 관한 것이다.

<6> CPU(중앙연산처리장치) 등에서의 슈퍼 스칼라(Superscalar) 파이프 라인 구조는, 한 클럭 사이클 동안에 동시에 여러 개의 명령어를 실행함으로써 CPU(중앙연산처리장치)

의 성능을 높이는 방법이다. 그러나, 슈퍼 스칼라(Superscalar) 파이프 라인 구조에서는 한 클럭 사이클 동안에 동시에 여러 개의 명령어가 실행될 수 있음에도 불구하고, 데이터 디펜던시(data dependency) 등의 문제로 인하여 시간상 명령어의 실행이 불가능한 구간, 즉, 웨이스트(waste)가 발생한다.

<7> 이에 반하여, 동시 다중 쓰레딩(SMT) 방법은 한 클럭 사이클 동안에 동시에 여러 쓰레드들(threads)이 동시에 존재하도록 하고, 쓰레드들(threads) 각각에 존재하는 명령어들이 동시에 실행되도록 하며, 특히 위와 같은 웨이스트(waste) 구간에서도 실행되도록 하여, 웨이스트(waste) 구간을 최소화함으로써 CPU(중앙연산처리장치)의 성능을 향상시키는 방법이다. 쓰레드(thread)는 한 프로세스 내의 여러 가지 서로 다른 컨트롤 포인트(control point)나 서로 다른 실행 경로(execution path)로 정의될 수 있고, 또는 서로 다른 프로그램으로도 정의될 수 있다. 따라서, 동시 다중 쓰레딩(SMT)은 여러 쓰레드들(threads) 각각의 명령어들이 동시에 수행되도록 하여 명령어 실행의 쓰루풋(throughput)을 높이는 방법이다.

<8> Dean M. Tullsen 등은 동시 다중 쓰레딩(SMT) 구조에서 여러 쓰레드들(threads) 중, 하나의 쓰레드(thread)를 선택하여 그 명령어를 페치(fetch)하는 몇 가지 알고리즘을 소개하였다. 이와 같이 쓰레드(thread)를 선택하고, 해당 명령어를 페치(fetch)하는 종래의 동시 다중 쓰레딩(SMT) 프로세서에 대해서는 미국 특허, "US6470443B1"에 잘 나타나 있다.

<9> 즉, 종래의 동시 다중 쓰레딩(SMT) 프로세서에서는, 새로운 쓰레드(thread)의 명령어 페치 시에, 레이턴시(latency)가 긴 명령어를 수행하는 쓰레드(thread) 선택을 피하기 위하여, "ICCOUNT", "BRCOUNT", "MISSCOUNT", 또는 "IQPOSN" 과 같은 방법을 사용한

다. 특히, 위와 같은 방법 중 "ICCOUNT"는 효과적인 방법으로서, 종래의 동시 다중 쓰레딩(SMT) 프로세서에 많이 채용되고 있다. "ICCOUNT" 방법은 명령어 디코딩, 명령어 재명명(rename), 명령어 지시(queue)와 같은 파이프 라인 단계에서, 가장 적은 수의 명령어를 가진 쓰레드(thread)를 선택하여 그 쓰레드(thread)의 명령어를 페치(fetch)하는 방법이다. 명령어 수가 적을수록 해당 쓰레드(thread)의 레이턴시(latency)가 작아지기 때문에, 이와 같은 방법은 효과적인 방법이라 할 수 있다. 이외에도, 파이프 라인 단계를 실행(execution) 단계까지 확장한 "ECOUNT"라는 방법도 있다.

<10> 이와 같이, "ICCOUNT" 방법 등을 사용하는 종래의 동시 다중 쓰레딩(SMT) 프로세서에서는, 단순히 쓰레드(thread)가 가지는 명령어 수만을 계산하여 쓰레드(thread)를 선택하고 명령어를 페치(fetch)한다. 그러나, 실제 레이턴시(latency)에 영향을 미치는 명령어들의 수행 주기 회수(cycle count)가 고려되지 않고, 단순히 명령어 수가 적은 쓰레드(thread)에 우선 순위를 부여하는 것은 CPU(중앙연산처리장치) 등의 프로세서 성능을 저하시키는 문제점이 있다. 예를 들어, 쓰레드(thread) A는 수행 주기 회수(cycle count)가 3인 곱셈 명령어 둘을 가지고, 쓰레드(thread) B는 수행 주기 회수(cycle count)가 1인 덧셈 명령어 네 개를 가질 경우, 쓰레드(thread) A의 수행이 쓰레드(thread) B의 수행보다 길어진다. 즉, 명령어의 개수는 오히려 적은 쓰레드(thread) A가 쓰레드(thread) B보다 명령어 지시(queue)를 더 오래 기다리므로, 큐 클록(queue clog)을 일으킬 수 있는 문제점이 있다.

【발명이 이루고자 하는 기술적 과제】

<11> 따라서, 본 발명이 이루고자하는 기술적 과제는, 명령어의 개수에 각 명령어의 수행 주기 회수(cycle count)를 가중치로 사용하여 동시 다중 쓰레딩(SMT)의 성능을 더욱 향상시킨 동시 다중 쓰레딩(SMT) 프로세서를 제공하는 데 있다.

<12> 본 발명이 이루고자하는 다른 기술적 과제는, 명령어의 개수에 각 명령어의 수행 주기 회수(cycle count)를 가중치로 사용하여 동시 다중 쓰레딩(SMT)의 성능을 더욱 향상시킨 동시 다중 쓰레딩(SMT) 프로세서의 동시 다중 쓰레딩(SMT) 방법을 제공하는 데 있다.

【발명의 구성 및 작용】

<13> 상기의 기술적 과제를 달성하기 위한 본 발명에 따른 동시 다중 쓰레딩(SMT) 프로세서는, 페치부(fetch unit), 명령어 캐쉬부(instruction cache), 명령어 디코딩부(instruction decoder), 레지스터 재명명부(register renamer), 명령어 지시부(instruction queue unit), 및 실행부(execution unit)를 구비한다.

<14> 상기 페치부(fetch unit)는 페치된 명령어(CIN)의 연산 종류 및 실행 지시된 명령어(QIS) 각각에 대응하는 쓰레드들(threads)에 대한 카운터 정보에 응답하여 쓰레드(thread)별로 카운팅을 수행하고, 쓰레드(thread)별 카운팅 값 중에서 가장 작은 카운팅 값에 대응하는 프로그램 카운터에서 어드레스(FAD)를 페치한다.

<15> 상기 명령어 캐쉬부(instruction cache)는 프로그램 카운터들 각각의 어드레스에 대응하는 명령어들을 저장하고, 상기 페치된 어드레스에 대응하는 상기 페치된 명령어(CIN)를 출력한다.

- <16> 상기 명령어 디코딩부(instruction decoder)는 상기 페치된 명령어(CIN)를 해석하여 상기 연산 종류 및 오퍼랜드로 사용될 레지스터의 어드레스(DAD)를 추출한다.
- <17> 상기 레지스터 재명명부(register renamer)는 데이터 디펜던시(data dependency)를 피할 수 있는 상기 레지스터 대용의 실제 레지스터의 어드레스를 재명명한다.
- <18> 상기 명령어 지시부(instruction queue unit)는 상기 재명명된 어드레스(RAD)에 응답하여 상기 페치된 명령어(CIN)에 대한 상기 실행 지시를 행한다.
- <19> 상기 실행부(execution unit)는 상기 실행 지시에 응답하여 상기 페치된 명령어(CIN)를 실행한다.
- <20> 상기 페치부(fetch unit)는, 디멀티플렉싱 제어기(demultiplexing controller), 싸이클 카운터들(cycle counters), 쓰레드 선택기(thread selector), 및 믹스(multiplexer)를 구비한다.
- <21> 상기 디멀티플렉싱 제어기(demultiplexing controller)는 상기 페치된 명령어(CIN)의 연산 종류 및 상기 실행 지시된 명령어(QIS) 각각에 대응하는 쓰레드들(threads)에 대한 상기 카운터 정보를 출력한다.
- <22> 상기 싸이클 카운터들(cycle counters)은 쓰레드(thread)별 상기 싸이클 카운터들(cycle counters)을 구비하며, 상기 카운터 정보에 대응하는 쓰레드(thread)에 할당되어 있는 상기 싸이클 카운터가 카운팅을 수행한다.
- <23> 상기 쓰레드 선택기(thread selector)는 상기 싸이클 카운터들(cycle counters)이 가리키는 수가 가장 작은 싸이클 카운터를 판별하고, 그에 대응하는 쓰레드(thread) 선택 정보(TSI)를 출력한다.

- <24> 상기 믹스(multiplexer)는 상기 스레드(thread) 선택 정보(TSI)에 응답하여 그에 대응하는 프로그램 카운터에서 어드레스(FAD)를 페치한다.
- <25> 상기 실행부(execution unit)는, 데이터 캐쉬(data cache), 레지스터들(registers), 및 기능 유닛들(functional units)을 구비한다.
- <26> 상기 데이터 캐쉬(data cache)는 상기 연산에 사용될 데이터들을 저장한다.
- <27> 상기 레지스터들(registers)은 상기 재명명된 어드레스(RAD)에 대응하는 다수개의 레지스터들(registers)을 구비한다.
- <28> 상기 기능 유닛들(functional units)은 이전 연산의 종료로 상기 실행 지시된 명령어(QIS) 수행에 사용될 오퍼랜드들이 생성되면, 상기 실행 지시에 응답하여 상기 페치된 명령어(CIN)를 실행한다.
- <29> 상기 카운터 정보는, 상기 페치된 명령어(CIN)의 연산 종류에 대응하는 수행 주기 회수(cycle count)를 합산하는 정보 및 상기 실행 지시된 명령어(QIS)에 대응하는 상기 수행 주기 회수(cycle count)를 감산하는 정보인 것을 특징으로 한다.
- <30> 상기 수행 주기 회수(cycle count)는, 상기 명령어 디코딩부(instruction decoder), 상기 레지스터 재명명부(register renamer), 및 상기 명령어 지시부(instruction queue unit)에서 소요되는 시스템 클럭의 사이클 카운트 값인 것을 특징으로 한다.
- <31> 상기의 다른 기술적 과제를 달성하기 위한 본 발명에 따른 동시 다중 스레딩(SMT) 프로세서의 동시 다중 스레딩(SMT) 방법은, 프로그램 카운터들 각각의 어드레스에 대응하는 명령어들을 저장하는 명령어 캐쉬부(instruction cache)를 구비하는 동시 다중 스

레딩(SMT) 프로세서의 동시 다중 쓰레딩(SMT) 방법에 있어서, 다음과 같은 단계를 구비한다.

<32> 즉, 본 발명에 따른 동시 다중 쓰레딩(SMT) 프로세서의 동시 다중 쓰레딩(SMT) 방법은, 먼저, 상기 동시 다중 쓰레딩(SMT) 프로세서에 구비되는 쓰레드(thread)별 싸이클 카운터들(cycle counters) 중에서, 폐치된 명령어(CIN)의 연산 종류 및 실행 지시된 명령어(QIS) 각각에 대응하는 쓰레드들(threads)에 대한 카운터 정보에 대응하는 쓰레드(thread)에 할당되어 있는 싸이클 카운터가 카운팅을 수행한다. 상기 동시 다중 쓰레딩(SMT) 프로세서는 상기 싸이클 카운터들(cycle counters)이 가리키는 수가 가장 작은 싸이클 카운터를 판별하여 그에 대응하는 프로그램 카운터에서 어드레스(FAD)를 폐치한다.

<33> 이에 따라, 상기 동시 다중 쓰레딩(SMT) 프로세서는 상기 명령어 캐쉬부(instruction cache)에서 상기 폐치된 어드레스에 대응하는 상기 폐치된 명령어(CIN)를 출력하고, 상기 폐치된 명령어(CIN)를 해석하여 상기 연산 종류 및 오퍼랜드로 사용될 레지스터의 어드레스(DAD)를 추출하며, 데이터 디펜던시(data dependency)를 피할 수 있는 상기 레지스터 대용의 실제 레지스터의 어드레스를 재명명한다.

<34> 다음에, 상기 동시 다중 쓰레딩(SMT) 프로세서는 상기 재명명된 어드레스(RAD)에 응답하여 상기 폐치된 명령어(CIN)에 대한 상기 실행 지시를 행하고, 상기 실행 지시에 응답하여 상기 폐치된 명령어(CIN)를 실행한다.

<35> 상기 프로그램 카운터에서 어드레스(FAD)의 폐치는, 다음과 같은 단계를 구비한다.

<36> 즉, 상기 프로그램 카운터에서 어드레스(FAD)의 페치에서는, 먼저, 상기 동시 다중 쓰레딩(SMT) 프로세서가, 상기 페치된 명령어(CIN)의 연산 종류 및 상기 실행 지시된 명령어(QIS) 각각에 대응하는 쓰레드들(threads)에 대한 상기 카운터 정보를 출력한다. 이에 따라, 상기 동시 다중 쓰레딩(SMT) 프로세서에 구비되는 쓰레드(thread)별 상기 싸이클 카운터들(cycle counters) 중에서, 상기 카운터 정보에 대응하는 쓰레드(thread)에 할당되어 있는 상기 싸이클 카운터가 카운팅을 수행한다. 다음에, 상기 동시 다중 쓰레딩(SMT) 프로세서는 상기 싸이클 카운터들(cycle counters)이 가리키는 수가 가장 작은 싸이클 카운터를 판별하고, 그에 대응하는 쓰레드(thread) 선택 정보(TSI)를 출력하며, 상기 쓰레드(thread) 선택 정보(TSI)에 응답하여 그에 대응하는 프로그램 카운터에서 어드레스(FAD)를 페치한다.

<37> 상기 실행은, 이전 연산의 종료로 상기 실행 지시된 명령어(QIS) 수행에 사용될 오퍼랜드들이 생성되면, 상기 동시 다중 쓰레딩(SMT) 프로세서에 구비되는 소정의 기능 유닛들(functional units)이 소정의 데이터 캐쉬(data cache) 및 레지스터들(registers)을 이용하여 상기 페치된 명령어(CIN)를 연산하는 것을 특징으로 한다.

<38> 상기 카운터 정보는, 상기 페치된 명령어(CIN)의 연산 종류에 대응하는 수행 주기 회수(cycle count)를 합산하는 정보 및 상기 실행 지시된 명령어(QIS)에 대응하는 상기 수행 주기 회수(cycle count)를 감산하는 정보인 것을 특징으로 한다.

<39> 상기 수행 주기 회수(cycle count)는, 상기 명령어 디코딩부(instruction decoder), 상기 레지스터 재명명부(register renamer), 및 상기 명령어 지시부(instruction queue unit)에서 소요되는 시스템 클럭의 사이클 카운트 값인 것을 특징으로 한다.

- <40> 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.
- <41> 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 각 도면에 제시된 동일한 참조부호는 동일한 부재를 나타낸다.
- <42> 도 1은 본 발명의 일실시예에 따른 동시 다중 쓰레딩(SMT) 프로세서의 블록도이다.
- <43> 도 1을 참조하면, 본 발명의 일실시예에 따른 동시 다중 쓰레딩(SMT) 프로세서는, 페치부(fetch unit)(110), 명령어 캐쉬부(instruction cache)(120), 명령어 디코딩부(instruction decoder)(130), 레지스터 재명명부(register renamer)(140), 명령어 지시부(instruction queue unit)(150), 및 실행부(execution unit)(160)를 구비한다.
- <44> 상기 페치부(fetch unit)(110)는 페치된 명령어(CIN)의 연산 종류 및 실행 지시된 명령어(QIS) 각각에 대응하는 쓰레드들(threads)에 대한 카운터 정보에 응답하여 쓰레드(thread)별로 카운팅을 수행하고, 쓰레드(thread)별 카운팅 값 중에서 가장 작은 카운팅 값에 대응하는 프로그램 카운터에서 어드레스(FAD)를 페치한다. 프로그램 카운터들은 쓰레드(thread)별로 하나씩 할당되며, 프로그램 카운터들 각각에서 출력되는 어드레스는, PC0 내지 PCN-1이다. 이때, PC0 내지 PCN-1 중 페치된 어드레스가 FAD에 해당한다.
- <45> 상기 페치된 명령어(CIN)를 해석하여 상기 연산 종류를 판별하는 것은 아래에서 기술할 상기 명령어 디코딩부(instruction decoder)(130)에서 이루어지고, 실행 지시된 명령어(QIS)에 대한 정보는 아래에서 기술할 상기 명령어 지시부(instruction queue unit)(150)로부터 피드백된다.

- <46> 여기서, 상기 카운터 정보는, 상기 폐치된 명령어(CIN)의 연산 종류에 대응하는 수행 주기 회수(cycle count)를 합산하는 정보 및 상기 실행 지시된 명령어(QIS)에 대응하는 상기 수행 주기 회수(cycle count)를 감산하는 정보이다. 상기 수행 주기 회수(cycle count)는, 상기 명령어 디코딩부(instruction decoder)(130), 상기 레지스터 재명명부(register renamer)(140), 및 상기 명령어 지시부(instruction queue unit)(150)에서 소요되는 시스템 클럭의 사이클 카운트 값이다.
- <47> 상기 명령어 캐쉬부(instruction cache)(120)는 프로그램 카운터들 각각의 어드레스에 대응하는 명령어들을 저장하고, 상기 폐치된 어드레스(FAD)에 대응하는 상기 폐치된 명령어(CIN)를 출력한다.
- <48> 상기 명령어 디코딩부(instruction decoder)(130)는 상기 폐치된 명령어(CIN)를 해석하여 상기 연산 종류 및 오퍼랜드로 사용될 레지스터의 어드레스(DAD)를 추출한다.
- <49> 상기 레지스터 재명명부(register renamer)(140)는 데이터 디펜던시(data dependency)를 피할 수 있는 상기 레지스터 대용의 실제 레지스터의 어드레스를 재명명한다. 이와 같이, 오퍼랜드로 사용될 레지스터에 대하여 그 대용의 실제 레지스터로 재명명하는 것은, 아키텍처상 정의된 레지스터보다 많은 양의 레지스터를 내부적으로 제공함으로써 데이터 디펜던시(data dependency)를 해결하기 위함이다.
- <50> 상기 명령어 지시부(instruction queue unit)(150)는 상기 재명명된 어드레스(RAD)에 응답하여 상기 폐치된 명령어(CIN)에 대한 상기 실행 지시를 행한다.
- <51> 상기 실행부(execution unit)(160)는 상기 실행 지시에 응답하여 상기 폐치된 명령어(CIN)를 실행한다. 이때, 실행은, 이전 연산의 종료로 상기 실행 지시된 명령어(QIS)

수행에 사용될 오퍼랜드들이 생성되면, 소정의 기능 유닛들(functional units)이, 소정의 데이터 캐쉬(data cache) 및 레지스터들(registers)을 이용하여 상기 실행 지시된 명령어(QIS)의 연산을 수행하는 것이다.

<52> 도 1을 참조하면, 상기 실행부(execution unit)(160)는, 데이터 캐쉬(data cache)(161), 레지스터들(registers)(163), 및 기능 유닛들(functional units)(165)을 구비한다.

<53> 상기 데이터 캐쉬(data cache)(161)는 상기 연산에 사용될 데이터들을 저장한다. 여기에 저장되는 데이터들은, 연산에 사용되는 여러 가지 필요한 소스 데이터들 또는 임시 데이터들로서, 동작 스피드의 향상을 위해 내부 캐쉬 메모리등에 저장되는 데이터들이다.

<54> 상기 레지스터들(registers)(163)은 상기 재명명된 어드레스(RAD)에 대응하는 다수 개의 레지스터들(registers)(163)을 구비한다. 여기의 레지스터들(registers)(163)은 상기 폐치된 명령어(CIN) 수행에서 오퍼랜드로 사용될 레지스터뿐만 아니라 연산 중에 사용되는 임시 저장 레지스터 또는 연산 결과를 저장하는 데스티네이션 레지스터(destination register) 등 연산에 필요한 모든 레지스터들(registers)(163)을 포함한다

<55> 상기 기능 유닛들(functional units)(165)은 이전 연산의 종료로 상기 실행 지시된 명령어(QIS) 수행에 사용될 오퍼랜드들이 생성되면, 상기 실행 지시에 응답하여 상기 폐치된 명령어(CIN)를 실행한다. 여기의 기능 유닛들(functional units)(165)은 합산기(adder), 승산기(multiplier), 쉬프터(shifter) 또는 축적기(accumulator) 등 기본 연산을 수행하는 블록들이다.

- <56> 도 2는 도 1의 페치부(fetch unit)(110)의 구체적인 블록도이다.
- <57> 도 2를 참조하면, 도 1의 페치부(fetch unit)(110)는, 디멀티플렉싱 제어기(demultiplexing controller)(111), 싸이클 카운터들(cycle counters)(113), 스레드 선택기(thread selector)(115), 및 믹스(multiplexer)(117)를 구비한다.
- <58> 상기 디멀티플렉싱 제어기(demultiplexing controller)(111)는 페치된 명령어(CIN)가 속하는 스레드(thread)를 판별하고, 명령어의 연산 종류에 대한 수행 주기 회수(cycle count)를 계산하여, 상기 페치된 명령어(CIN)의 연산 종류 및 상기 실행 지시된 명령어(QIS) 각각에 대응하는 스레드(thread)에 대한 상기 카운터 정보를 출력한다. 위에서 기술한 바와 같이, 상기 페치된 명령어(CIN)를 해석하여 상기 연산 종류를 판별하는 것은, 상기 명령어 디코딩부(instruction decoder)(130)에서 이루어지고, 실행 지시된 명령어(QIS)에 대한 정보는 상기 명령어 지시부(instruction queue unit)(150)로부터 피드백된다.
- <59> 여기서도, 카운터 정보는, 상기 페치된 명령어(CIN)의 연산 종류에 대응하는 수행 주기 회수(cycle count)를 합산하는 정보 및 상기 실행 지시된 명령어(QIS)에 대응하는 상기 수행 주기 회수(cycle count)를 감산하는 정보이다. 상기 수행 주기 회수(cycle count)는, 상기 명령어 디코딩부(instruction decoder)(130), 상기 레지스터 재명명부(register renamer)(140), 및 상기 명령어 지시부(instruction queue unit)(150)에서 소요되는 시스템 클럭의 사이클 카운트 값이다.
- <60> 상기 싸이클 카운터들(cycle counters)(113)은 스레드(thread)별 상기 싸이클 카운터들(cycle counters)(113)을 구비하며, 상기 카운터 정보에 대응하는 스레드(thread)에 할당되어 있는 상기 싸이클 카운터가 카운팅을 수행한다.

- <61> 상기 쓰레드 선택기(thread selector)(115)는 상기 싸이클 카운터들(cycle counters)(113)이 가리키는 수가 가장 작은 싸이클 카운터를 판별하고, 그에 대응하는 쓰레드(thread) 선택 정보(TSI)를 출력한다.
- <62> 상기 믹스(multiplexer)(117)는 상기 쓰레드(thread) 선택 정보(TSI)에 응답하여 그에 대응하는 프로그램 카운터에서 어드레스(FAD)를 페치한다. 프로그램 카운터들은 쓰레드(thread)별로 하나씩 할당되며, 프로그램 카운터들 각각에서 출력되는 어드레스는, PC0 내지 PCN-1이다. 이때, PC0 내지 PCN-1 중 페치된 어드레스가 FAD에 해당한다.
- <63> 상기한 바와 같은, 본 발명의 일실시예에 따른 동시 다중 쓰레딩(SMT) 프로세서의 동작을 좀더 상세하게 설명한다.
- <64> 도 3은 도 2의 동시 다중 쓰레딩(SMT) 프로세서의 동작 설명을 위한 흐름도이다.
- <65> 도 3을 참조하면, 도 2의 동시 다중 쓰레딩(SMT) 프로세서의 동시 다중 쓰레딩(SMT) 방법은, 프로그램 카운터들 각각의 어드레스에 대응하는 명령어들을 저장하는 명령어 캐쉬부(instruction cache)(120)를 구비하는 동시 다중 쓰레딩(SMT) 프로세서의 동시 다중 쓰레딩(SMT) 방법에 있어서, 다음과 같은 단계를 구비한다.
- <66> 즉, 본 발명에 따른 동시 다중 쓰레딩(SMT) 프로세서의 동시 다중 쓰레딩(SMT) 방법은, 먼저, 페치부(fetch unit)(110)가 명령어 수에 그 수행 주기 회수(cycle count)를 가중치로 사용하여 쓰레드(thread)를 선택하고, 그 쓰레드(thread) 중 실행될 명령어의 어드레스를 가리키고 있는 프로그램 카운터에서 어드레스(FAD)를 페치한다(S310).
- <67> 즉, 페치부(fetch unit)(110)가 프로그램 카운터에서 어드레스(FAD)를 페치하는 과정을 좀더 자세하게 말하면, 다음과 같다. 먼저, 페치부(fetch unit)(110)에 구비된 디

멀티플렉싱 제어기(demultiplexing controller)(111)가, 페치된 명령어(CIN)가 속하는 쓰레드(thread)를 판별하고, 명령어의 연산 종류에 대한 수행 주기 회수(cycle count)를 계산하여, 페치된 명령어(CIN)의 연산 종류 및 실행 지시된 명령어(QIS) 각각에 대응하는 쓰레드(thread)에 대한 카운터 정보를 출력한다. 이에 따라, 페치부(fetch unit)(110)에 구비되는 쓰레드(thread)별 싸이클 카운터들(cycle counters)(113) 중에서, 상기 카운터 정보에 대응하는 쓰레드(thread)에 할당되어 있는 싸이클 카운터가 카운팅을 수행한다.

<68> 다음에, 페치부(fetch unit)(110)에 구비되는 상기 쓰레드 선택기(thread selector)(115)는 상기 싸이클 카운터들(cycle counters)(113)이 가리키는 수가 가장 작은 싸이클 카운터를 판별하고, 그에 대응하는 쓰레드(thread) 선택 정보(TSI)를 출력하며, 페치부(fetch unit)(110)에 구비되는 상기 믹스(multiplexer)(117)는 상기 쓰레드(thread) 선택 정보(TSI)에 응답하여 그에 대응하는 프로그램 카운터에서 어드레스(FAD)를 페치한다(S310). 어느 한 쓰레드(thread)에 속하는 최초의 명령어가 페치될 때, 그 쓰레드(thread)에 대응하는 상기 싸이클 카운터는 상기 디멀티플렉싱 제어기(demultiplexing controller)(111)에 의하여 리셋되어 다시 카운팅을 시작한다.

<69> 위에서 기술한 바와 같이, 상기 페치된 명령어(CIN)를 해석하여 상기 연산 종류를 판별하는 것은, 상기 명령어 디코딩부(instruction decoder)(130)에서 이루어지고, 실행 지시된 명령어(QIS)에 대한 정보는 상기 명령어 지시부(instruction queue unit)(150)로부터 피드백된다. 이때, 상기 카운터 정보는, 상기 페치된 명령어(CIN)의 연산 종류에 대응하는 수행 주기 회수(cycle count)를 합산하는 정보 및 상기 실행 지시된 명령어(QIS)에 대응하는 상기 수행 주기 회수(cycle count)를 감산하는 정보이다(S370~S380).

상기 수행 주기 회수(cycle count)는, 상기 명령어 디코딩부(instruction decoder)(130), 상기 레지스터 재명명부(register renamer)(140), 및 상기 명령어 지시부(instruction queue unit)(150)에서 소요되는 시스템 클럭의 사이클 카운트 값이다.

<70> 한편, 상기 명령어 캐쉬부(instruction cache)(120)에서 상기 폐치된 어드레스에 대응하는 상기 폐치된 명령어(CIN)를 출력하면, 상기 명령어 디코딩부(instruction decoder)(130)는 상기 폐치된 명령어(CIN)를 해석하여 상기 연산 종류 및 오퍼랜드로 사용될 레지스터의 어드레스(DAD)를 추출하고(S320), 상기 레지스터 재명명부(register renamer)(140)는 데이터 디펜던시(data dependency)를 피할 수 있는 상기 레지스터 대용의 실제 레지스터의 어드레스를 재명명한다(S330).

<71> 다음에, 상기 명령어 지시부(instruction queue unit)(150)는 상기 재명명된 어드레스(RAD)에 응답하여 상기 폐치된 명령어(CIN)에 대한 상기 실행 지시를 행하고(S340), 상기 실행부(execution unit)(160)가 상기 실행 지시에 응답하여 상기 폐치된 명령어(CIN)를 실행한다(S360). 이와 같은 명령어 실행에 따라 연산된 결과가 데스티네이션 레지스터(destination register)에 저장된다(S360).

<72> 여기서, 상기 실행은, 이전 연산의 종료로 상기 실행 지시된 명령어(QIS) 수행에 사용될 오퍼랜드들이 생성되면, 상기 실행부(execution unit)(160)에 구비되는 소정의 기능 유닛들(functional units)(165)이 소정의 데이터 캐쉬(data cache)(161) 및 레지스터들(registers)(163)을 이용하여 상기 폐치된 명령어(CIN)를 연산하는 것이다(S350).

<73> 위에서 기술한 바와 같이, 본 발명의 일실시예에 따른 동시 다중 쓰레딩(SMT) 프로세서는, 상기 폐치부(fetch unit)(110)가 명령어 수에 그 수행 주기 회수(cycle count)

를 가중치로 사용하여 스레드(thread)를 선택하고, 그 스레드(thread) 중 실행될 명령어의 어드레스를 가리키고 있는 프로그램 카운터에서 어드레스(FAD)를 폐치한다.

<74> 즉, 폐치부(fetch unit)(110)는 폐치된 명령어(CIN)가 속하는 스레드(thread)를 판별하고, 명령어의 연산 종류에 대한 수행 주기 회수(cycle count)를 계산하여, 폐치된 명령어(CIN)의 연산 종류 및 실행 지시된 명령어(QIS) 각각에 대응하는 스레드(thread)에 대한 카운터 정보를 출력한다. 폐치부(fetch unit)(110)에 구비되는 스레드(thread)별 싸이클 카운터들(cycle counters)(113) 중에서, 상기 카운터 정보에 대응하는 스레드(thread)에 할당되어 있는 싸이클 카운터는 카운팅을 수행한다.

<75> 다음에, 폐치부(fetch unit)(110)는 상기 싸이클 카운터들(cycle counters)(113)이 가리키는 수가 가장 작은 싸이클 카운터를 판별하고, 그에 대응하는 스레드(thread) 선택 정보(TSI)를 출력한다. 이에 따라 폐치부(fetch unit)(110)에 구비되는 상기 믹스(multiplexer)(117)는 상기 스레드(thread) 선택 정보(TSI)에 응답하여 그에 대응하는 프로그램 카운터에서 어드레스(FAD)를 폐치한다.

<76> 이상에서와 같이 도면과 명세서에서 최적 실시예가 개시되었다. 여기서 특정한 용어들이 사용되었으나, 이는 단지 본 발명을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 발명의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의해 정해져야 할 것이다.

【발명의 효과】

<77> 상술한 바와 같이 본 발명에 따른 동시 다중 쓰레딩(SMT) 프로세서는, 명령어의 개수에 각 명령어의 수행 주기 회수(cycle count)를 가중치로 사용하여 쓰레드(thread)를 선택하고 그 쓰레드(thread)의 명령어를 페치(fetch)하므로, SMT의 성능을 더욱 향상시키고, 결국 CPU(중앙연산처리장치) 등의 시스템 성능을 더욱 향상시키는 효과가 있다.

【특허청구범위】**【청구항 1】**

폐치된 명령어의 연산 종류 및 실행 지시된 명령어 각각에 대응하는 쓰레드들에 대한 카운터 정보에 응답하여 쓰레드별로 카운팅을 수행하고, 쓰레드별 카운팅 값 중에서 가장 작은 카운팅 값에 대응하는 프로그램 카운터에서 어드레스를 폐치하는 폐치부;

프로그램 카운터들 각각의 어드레스에 대응하는 명령어들을 저장하고, 상기 폐치된 어드레스에 대응하는 상기 폐치된 명령어를 출력하는 명령어 캐쉬부;

상기 폐치된 명령어를 해석하여 상기 연산 종류 및 오퍼랜드로 사용될 레지스터의 어드레스를 추출하는 명령어 디코딩부;

데이터 디펜던시를 피할 수 있는 상기 레지스터 대용의 실제 레지스터의 어드레스를 재명명하는 레지스터 재명명부;

상기 재명명된 어드레스에 응답하여 상기 폐치된 명령어에 대한 상기 실행 지시를 행하는 명령어 지시부; 및

상기 실행 지시에 응답하여 상기 폐치된 명령어를 실행하는 실행부를 구비하는 것을 특징으로 하는 동시 다중 쓰레딩 프로세서.

【청구항 2】

제 1항에 있어서, 상기 폐치부는,

상기 폐치된 명령어의 연산 종류 및 상기 실행 지시된 명령어 각각에 대응하는 쓰레드들에 대한 상기 카운터 정보를 출력하는 디멀티플렉싱 제어기;

쓰레드별 상기 싸이클 카운터들을 구비하며, 상기 카운터 정보에 대응하는 쓰레드에 할당되어 있는 상기 싸이클 카운터가 카운팅을 수행하는 싸이클 카운터들;

상기 싸이클 카운터들이 가리키는 수가 가장 작은 싸이클 카운터를 판별하고, 그에 대응하는 쓰레드 선택 정보를 출력하는 쓰레드 선택기; 및

상기 쓰레드 선택 정보에 응답하여 그에 대응하는 프로그램 카운터에서 어드레스를 폐치하는 맥스를 구비하는 것을 특징으로 하는 동시 다중 쓰레딩 프로세서.

【청구항 3】

제 1항에 있어서, 상기 실행부는,

상기 연산에 사용될 데이터들을 저장하는 데이터 캐쉬;

상기 재명명된 어드레스에 대응하는 다수개의 레지스터들을 구비하는 레지스터들;

및

이전 연산의 종료로 상기 실행 지시된 명령어 수행에 사용될 오퍼랜드들이 생성되면, 상기 실행 지시에 응답하여 상기 폐치된 명령어를 실행하는 기능 유닛들을 구비하는 것을 특징으로 하는 동시 다중 쓰레딩 프로세서.

【청구항 4】

제 1항 또는 제 2항에 있어서, 상기 카운터 정보는,

상기 폐치된 명령어의 연산 종류에 대응하는 수행 주기 회수를 합산하는 정보 및 상기 실행 지시된 명령어에 대응하는 상기 수행 주기 회수를 감산하는 정보인 것을 특징으로 하는 동시 다중 쓰레딩 프로세서.

【청구항 5】

제 4항에 있어서, 상기 수행 주기 회수는,

상기 명령어 디코딩부, 상기 레지스터 재명명부, 및 상기 명령어 지시부에서 소요되는 시스템 클럭의 사이클 카운트 값인 것을 특징으로 하는 동시 다중 쓰레딩 프로세서.

【청구항 6】

프로그램 카운터들 각각의 어드레스에 대응하는 명령어들을 저장하는 명령어 캐쉬부를 구비하는 동시 다중 쓰레딩 프로세서의 동시 다중 쓰레딩 방법에 있어서,

상기 동시 다중 쓰레딩 프로세서에 구비되는 쓰레드별 사이클 카운터들 중에서, 폐치된 명령어의 연산 종류 및 실행 지시된 명령어 각각에 대응하는 쓰레드들에 대한 카운터 정보에 대응하는 쓰레드에 할당되어 있는 사이클 카운터가 카운팅을 수행하는 단계;

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 사이클 카운터들이 가리키는 수가 가장 작은 사이클 카운터를 판별하여 그에 대응하는 프로그램 카운터에서 어드레스를 폐치하는 단계;

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 명령어 캐쉬부에서 상기 폐치된 어드레스에 대응하는 상기 폐치된 명령어를 출력하는 단계;

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 폐치된 명령어를 해석하여 상기 연산 종류 및 오퍼랜드로 사용될 레지스터의 어드레스를 추출하는 단계;



상기 동시 다중 쓰레딩 프로세서에 의하여, 데이터 디펜던시를 피할 수 있는 상기 레지스터 대용의 실제 레지스터의 어드레스를 재명명하는 단계;

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 재명명된 어드레스에 응답하여 상기 폐치된 명령어에 대한 상기 실행 지시를 행하는 단계; 및

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 실행 지시에 응답하여 상기 폐치된 명령어를 실행하는 단계를 구비하는 것을 특징으로 하는 동시 다중 쓰레딩 프로세서의 동시 다중 쓰레딩 방법.

【청구항 7】

제 6항에 있어서, 상기 프로그램 카운터에서 어드레스의 폐치는,

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 폐치된 명령어의 연산 종류 및 상기 실행 지시된 명령어 각각에 대응하는 쓰레드들에 대한 상기 카운터 정보를 출력하는 단계;

상기 동시 다중 쓰레딩 프로세서에 구비되는 쓰레드별 상기 싸이클 카운터들 중에서, 상기 카운터 정보에 대응하는 쓰레드에 할당되어 있는 상기 싸이클 카운터가 카운팅을 수행하는 단계;

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 싸이클 카운터들이 가리키는 수가 가장 작은 싸이클 카운터를 판별하고, 그에 대응하는 쓰레드 선택 정보를 출력하는 단계; 및



상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 쓰레드 선택 정보에 응답하여 그에 대응하는 프로그램 카운터에서 어드레스를 폐치하는 단계를 구비하는 것을 특징으로 하는 동시 다중 쓰레딩 프로세서의 동시 다중 쓰레딩 방법.

【청구항 8】

제 6항에 있어서, 상기 실행은,

상기 동시 다중 쓰레딩 프로세서에 구비되는 소정의 기능 유닛들에 의하여, 이전 연산의 종료로 상기 실행 지시된 명령어 수행에 사용될 오퍼랜드들이 생성되면, 소정의 데이터 캐쉬 및 레지스터들을 이용하여 상기 폐치된 명령어를 연산하는 것을 특징으로 하는 동시 다중 쓰레딩 프로세서의 동시 다중 쓰레딩 방법.

【청구항 9】

제 6항 또는 제 7항에 있어서, 상기 카운터 정보는,

상기 폐치된 명령어의 연산 종류에 대응하는 수행 주기 회수를 합산하는 정보 및 상기 실행 지시된 명령어에 대응하는 상기 수행 주기 회수를 감산하는 정보인 것을 특징으로 하는 동시 다중 쓰레딩 프로세서의 동시 다중 쓰레딩 방법.

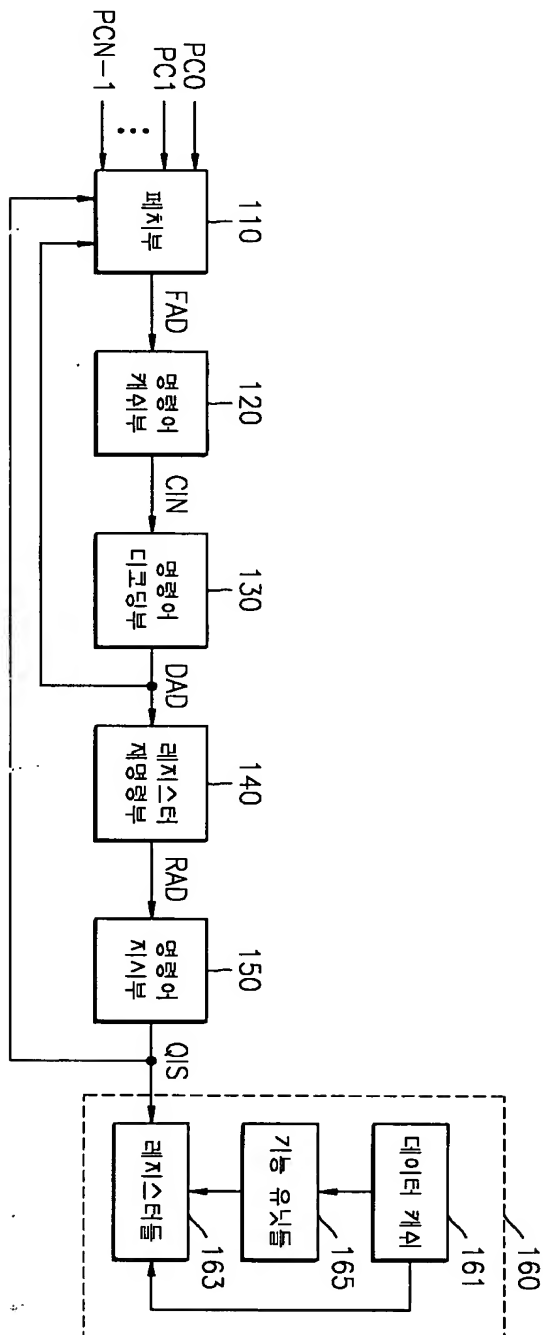
【청구항 10】

제 9항에 있어서, 상기 수행 주기 회수는,

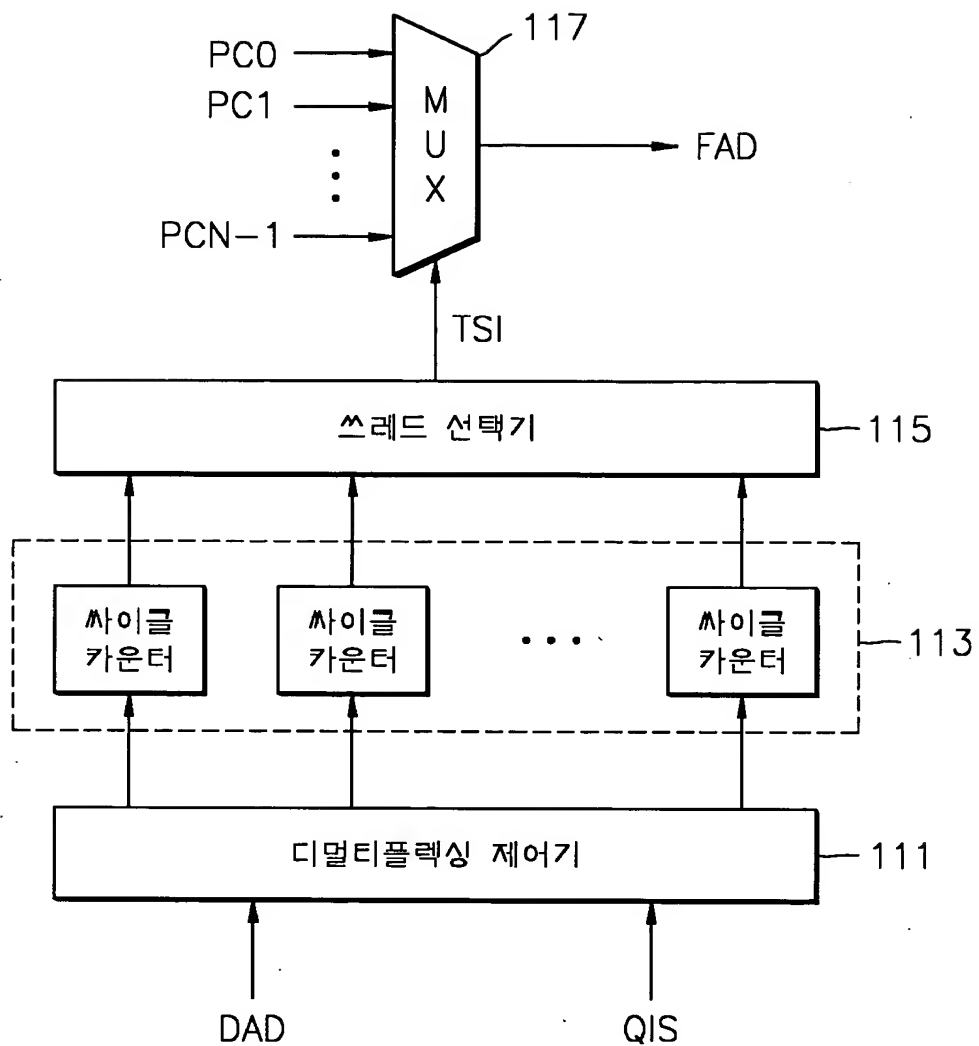
상기 명령어 디코딩부, 상기 레지스터 재명명부, 및 상기 명령어 지시부에서 소요되는 시스템 클럭의 사이클 카운트 값인 것을 특징으로 하는 동시 다중 쓰레딩 프로세서의 동시 다중 쓰레딩 방법.

【도면】

【도 1】



【도 2】



【도 3】

